

## **An Approach to Integrating Learning and Engagement Strategies (LESs) into CS Class Activities**

### **Dr. Peter J. Clarke, Florida International University**

Peter J. Clarke received his B.Sc. degree in Computer Science and Mathematics from the University of the West Indies (Cave Hill) in 1987, M.S. degree from SUNY Binghamton University in 1996 and Ph.D. in Computer Science from Clemson University in 2003. His research interests are in the areas of software testing, software metrics, model-driven software development, domain-specific modeling languages, and computer science education. He is currently an associate professor in the School of Computing and Information Sciences at Florida International University. He is a member of the ACM (SIGSOFT, SIGCSE, and SIGAPP); IEEE Computer Society; and a member of the Association for Software Testing (AST).

### **Dr. Debra Lee Davis, Florida International University**

Dr. Debra Davis is an Instructor in the School of Computing and Information Sciences at Florida International University. Her research interests emphasize interdisciplinary topics including understanding and improving: (1) Computer Science education, including increasing participation of women; (2) educational applications and techniques for online STEM learning; and (3) complex human-machine interactions. She has a Ph.D. and M.A., in Cognitive Developmental Psychology from the University of Texas at Austin, and an M.S. in Computer Science from FIU.

### **Dr. Ingrid A. Buckley, Florida Gulf Coast University**

Dr. Ingrid Buckley is an Assistant Professor in the Software Engineering Department at Florida Gulf Coast University. She holds a PhD (2012) in Computer Science from Florida Atlantic University. Dr. Buckley's research interests include software engineering education, fault tolerant system design, cybersecurity and secure software development. She has authored several peer-reviewed conference and journal papers. She has received grants from the National Science Foundation (NSF) and Cyber Florida. She collaborates with researchers at other universities nationwide, specifically in the areas of software engineering and cybersecurity pedagogy. Dr. Buckley is a member of the Upsilon Pi Epsilon, Golden Key International and the American Society for Engineering Education (ASEE).

### **Dr. Geoff Potvin, Florida International University**

### **Dr. Mandayam Thirunarayanan, Florida International University**

Mandayam Osuri Thirunarayanan is an associate professor in the School of Education and Human Development (SEHD), in the College of Arts, Sciences, and Education (CASE) at Florida International University, in Miami, Florida. He teaches courses in learning technologies at the undergraduate, masters and doctoral levels. He also supervises doctoral dissertations. His research interests include distance education, cyber-learning, and videogames in education. He has proposed innovative and futuristic educational applications.

### **Prof. Edward L. Jones, Florida A&M University/Florida State University**

Edward L. Jones is Professor and Interim Chairman of the CIS Department at Florida A&M University (FAMU) in Tallahassee, Florida. He joined the faculty after 13 years as a software engineer at Harris Corporation. His SIGCSE 2001 paper, "Integrating testing into the curriculum - arsenic in small doses," proposed a vision for incorporating testing into the CS curriculum. He founded the FAMU Software TestLab, funded by federal and corporate grants for training software testing practitioners and researchers. He has directed master's theses in software testing and test automation. Born in North Carolina, he completed the B.S., summa cum laude, in Math from Johnson C. Smith University, the M.S. in Computer Science from Cornell, and the Ph.D. in Computer Science from the University of North Carolina at Chapel Hill, in 1984. His research interests are software testing, technology transfer and CS education.

# An Approach to Integrating Learning and Engagement Strategies (LESS) into CS Class Activities

## Abstract

The recent surge in Computer Science (CS), Information Technology (IT) and Software Engineering (SE) majors at universities across the nation has resulted in a number of challenges, including mitigating students' lack of preparedness for these programs. To overcome some of these challenges it is imperative that CS/IT/SE instructors make use of evidence-based pedagogical techniques that improve student learning and engagement in the classroom, e.g., active learning. In this paper we present a model that uses learning and engagement strategies (LESS) to improve student learning in face-to-face (F2F) classes with the support of a cyberlearning environment. These LESSs include *collaborative learning*, *gamification*, *problem-based learning* and *social interaction*. SEP-CyLE (*Software Engineering and Programming Cyberlearning Environment*) contains vetted learning content in the form of learning objects and tutorials, and can be configured to use various combinations of LESSs. We describe how LESSs are integrated into the F2F and online activities of a software testing class. The results of a study we conducted that integrates LESSs into the F2F activities of a software testing class show the potential positive impact such a pedagogical approach can have on CS/IT/SE classes.

## 1 Introduction

Due to the ubiquitous nature of computing in the 21st century, there is a great demand for a workforce that is knowledgeable in the areas of Computer Science (CS), Information technology (IT) and Software Engineering (SE). This fact is reflected in the large number of jobs expected in CS/IT/SE in the coming years. As a result, there has been a dramatic increase in the number of students enrolling in CS/IT/SE programs across the nation [1]. This increase in enrollment has several challenges associated with it, including: classroom space, sufficient faculty/instructors and demand for lab space, among others [1]. One challenge that still needs further investigation is the declining student performance possibly due to the large number of students entering programs who may not be well prepared to undertake courses in CS/IT/SE. This specific challenge is further compounded by the fact that most academic institutions are being rewarded based on performance metrics such as 4-year graduation rates.

To overcome the challenge related to a potential decline in student performance, it is imperative that CS/IT/SE instructors make use of those evidence-based pedagogical approaches that improve student learning in the classroom. Several of these approaches have started to filter into CS/IT/SE education, including active learning [2] which includes problem-based learning [3] and collaborative learning [4], gamification [5], and social interaction [6], among others. Although there has been some use of these approaches in the CS/IT/SE classroom, little research has been done on how combinations of these approaches impact student learning and engagement. We refer to *collaborative learning*, *gamification*, *problem-based learning* and *social interaction* as *learning and engagement strategies* (LESS).

In this paper, we present a model that is intended to motivate research into improving student learning by integrating combinations of LESs into face-to-face (F2F) and online learning activities. During the past four years the authors of this paper have been involved in using LESs in a cyberlearning environment to determine the impact on student learning in CS/IT/SE courses. This cyberlearning environment, initially known as WReSTT - *Web-based Repository of Software Testing Tutorials* (now SEP-CyLE - *Software Engineering and Programming Cyberlearning Environment* [7]), has shown that using LESs in a cyberlearning environment can improve student learning in the practical aspects of software testing [8, 9]. Unlike previous works by the authors, this paper presents their experiences of using LESs in the F2F activities of a software testing class.

The main contributions of this work are as follows:

- A proposed model that can be used by instructors and researchers to determine which combinations of LESs and the context of how LESs are used in F2F and online activities can best improve student learning and engagement.
- A description of how LESs may be used in a CS class, specifically a software testing class.
- Initial results showing the impact of using LESs in F2F class activities.

The remainder of the paper is organized as follows. Section 2 provides a literature review on LESs and how they are used in software engineering classes. Section 3 describes the pedagogical model that uses LESs in the classroom. Section 4 describes our study that uses LESs in the classroom and we conclude in Section 5.

## 2 Literature Review

In this section we provide a literature review of learning and engagement strategies (LESs) and how LESs have been used in software engineering classes. There are many evidence-based pedagogical techniques that have been used to improve student learning in engineering and computer science education [2, 10]. In this work, we focus on the following LESs: *collaborative learning*, *gamification*, *problem-based learning* and *social interaction*. It should be noted that these LESs do not form a disjointed set of pedagogical techniques, however we attempt to identify differences between these techniques by describing how they are used in a software testing class, see Section 3.4.

### 2.1 Learning and Engagement Strategies (LESs)

*Collaborative learning* is where two or more people work in groups mutually searching for understanding, solutions, or meanings, or creating a product [11]. Smith et al. [11] state that there is a wide variation in the collaborative learning activities, but they must be centered on the students' exploration or application of the material. Collaborative learning promotes several education goals, including: *involvement* - students participate more in their learning, interact with other students and interact with the teachers; *cooperation and teamwork* - students recognize different views and work together to build consensus to resolve these differences; and *civic responsibility* - encourages students to have a voice in shaping the ideas and values [11].

*Gamification* uses game design elements and game mechanics to improve user experience and engagement with a system [12], which can be applied to an educational context [5]. Importantly,

gamification is not about adding “games” to a system. It is about taking elements commonly implemented in games that humans find fun and enjoyable, and applying those to other domains in a manner that promotes peoples motivation for engaging in desired behaviors and increasing engagement. Common game design elements include: goals/challenges, rapid feedback, leader boards, a points system and levels, among others [5].

*Problem-based learning* (PBL) is an approach to learning and instruction in which students tackle problems in small groups under the supervision of a tutor [13]. The problems are presented as a set of events (significant or otherwise) that are associated with reality. These events are analyzed and discussed by the group in term of the fundamental principles, mechanisms and processes. PBL has the potential to significantly increase student performance in the classroom by: (a) activating prior knowledge; (b) elaborating on prior knowledge through discussion; (c) restructuring of knowledge; (d) learning in context; and (e) engaging in open-ended discussion thereby increasing the student’s curiosity [13].

*Social Interaction* is an approach that enhances knowledge acquisition through social activities, such as students establishing meaningful dialogue within student groups and with teachers [6]. Interacting with other people helps a learner to organize their thoughts, improve their understanding by reflection and identify gaps in their reasoning [14]. There are several variants of social interaction and learning, including peer learning, reciprocal teaching, learning by teaching and learning by observation, among others [14]. The impact of social interaction on learning can be examined using various collaborative learning approaches.

## 2.2 Related Work

Cattaneo [2] classifies five active learning pedagogies using constructivist elements through theoretical and practical lenses, thereby providing researchers and practitioners with a useful tool to compare these pedagogies. Although we focus on PBL in our work there are other active learning pedagogies that can also be used in place of PBL. Gehringer [4] presents a survey of dozens of active and collaborative learning strategies (ACL) that have been used in programming. The paper describes how different specific learning activities incorporate LESs, although the term LESs is not used, into teaching programming.

Liccardi et al. [15] investigated the role of social networks in computer science education. It was found that social interaction within an online framework can help university students share experiences and collaborate on relevant topics. As such, social networks can act as a pedagogical agent, for example, with problem-based learning. Junhua [16] performs a study in software engineering classes that compares lecture-homework-project teaching approaches to peer-to-peer active learning when combined with design-based learning approaches. Although both approaches show that student do show improved performance, the peer-to-peer active learning and design-based learning approach received much greater interest, engagement, and intrinsic motivation. Dicheva et al. [5] present a survey of published empirical results on the applications of gamification in education. The authors further stated that although most of the papers surveyed showed promising results there is still more research needed to determine the impact on student learning.

Clarke et al. [9] describe how WReSTT (Web-Based Repository of Software Testing Tutorials) is used to supplement the learning materials in a software testing class. The study presented by

Clarke et al. spanned 3 semesters and the results showed that students found WReSTT to be a useful resource for learning about software testing techniques and testing tools. The version of WReSTT used in the study included collaborative learning, gamification and social interaction. The study presented in [9] was conducted during Fall 2011, Fall 2012, and Fall 2013, around the same time LESs were being integrated into the F2F activities of the software testing class.

The work presented in this section shows how different LESs may be used in software engineering classes to improve student learning. Our work differs from these works by proposing a model that can be used to integrate LESs into F2F and online class activities, and evaluate which combination(s) of LESs can be most effective on improving student learning.

### 3 Pedagogical Approach using LESs

In this section we present our proposed LES integration model (LESIM), a brief overview of SEP-CyLE, and a description of how LESs are integrated into the F2F and online activities of a software testing class. Preliminary results obtained by comparing the midterm exam scores prior to using LESs and while using LESs in the classroom are also presented.

#### 3.1 LES Integration Model (LESIM)

Conceptually LESIM is used to improve student learning based on the model shown in Figure 1. The top of the figure shows the pedagogical approaches (LESs and traditional approach) that are integrated into F2F and online class activities, and the F2F and online learning content. The traditional approach mainly involves lecture style teaching and completing online assignments (no LESs used). By combining the LESs and traditional approaches to learning in different ways we can maximize student learning and engagement, and possibly tailor this combination to different types of learners. The model presented in Figure 1 is motivated by three components used in cognitive load theory [17, 18], which are *intrinsic load* - the content being learned, *extraneous load* - the load which interferes with the learning process, and *germane load* - concerned with schema development. In general, it is desirable to mitigate extraneous load [19] so that more opportunity for schema construction (germane load) is possible. Note that although our model currently shows only 4 LESs, we expect that this number will increase over time as more pedagogical approaches are integrated.

The main objective of the proposed model is through research find the values of  $a, b, c, d, e, q, r, s,$  and  $t$  in the equations F2FA ( $a\text{CL} + b\text{GA} + c\text{PBL} + d\text{SI} + e\text{LS}$ ) and OA ( $q\text{CL} + r\text{GA} + s\text{PBL} + t\text{SI}$ ) that maximizes student learning and engagement. Where F2FA and OA are the face-to-face and online activities respectively. **CL** - collaborative learning, **GA** - gamification, **PBL** - problem-based learning and **SI** - social interaction are the LESs, and **LS** - lecture style the traditional approach. You may think of  $a, b, c, d, e, q, r, s,$  and  $t$  as the percentage of the respective pedagogical approach being used. For F2F classes these variables could represent class time. It is expected that along with the percentages there will be a context in which the LESs are used. For example, **CL** in the context of F2F activities (F2FA) will most probably be different to the **CL** in the context of the online activities (OA), see Section 3.4 for additional details.

Basing our model in the context of cognitive load theory [17, 18] our objective is to improve a student's implicit knowledge on a topic, to this end using a LES in the wrong context can do more

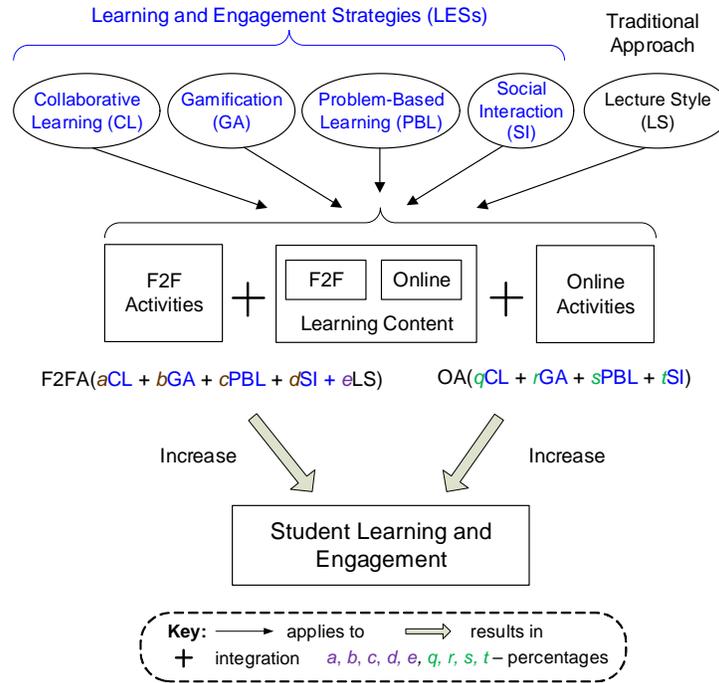


Figure 1: LES integration model (LESIM).

harm than good. For example, when using GA in a class activity that entails a complex game with many players and levels (extraneous load), students may gain little or no knowledge (germane load) after the game is over. On the other side, to improve student learning some LESs may need to be used in concert with each other. For example, when students solve problems in small groups the LESs involved are usually CL, PBL and SI. It should be noted that the LES integration model shown in Figure 1 is *extensible* and *adaptable*.

### 3.2 Cyberlearning Environment

The authors of this paper are a part of a larger study investigating the impact of using different combinations of LESs in a cyberlearning environment (SEP-CyLE - *Software Engineering and Programming Cyberlearning Environment*) [7]. SEP-CyLE provides students and instructors with vetted digital learning content and uses various embedded LESs to support pedagogy. The learning content is provided in the form of Digital Learning Objects (DLOs) [20] on various Software Engineering and Programming (SEP) topics, and tutorials for several SEP tools. DLOs are 5 to 10-minute chunks of critical learning content and have a practice assessment and a graded assessment. Unlike the DLOs the tutorials do not have an assessment component.

In SEP-CyLE the embedded LESs are implemented as follows: CL - uses virtual teams to collaborate to complete online assignments, posting comments on the work of other teams and sharing information that is helpful to the entire class via the class forum. GA - uses virtual points, a leader board, and the allocations of points based on various activities, such as completing assignments and posting to a forum, among others. SI - includes the use of user profiles, message forums, and posting comments on learning content, among others. Note that the LESs in the context of the online activities are slightly different from the F2F activities. We are still in the process of integrating the PBL which will involve solving problems using a virtual problem-based environment.

SEP-CyLE is an instance of STEM-CyLE<sup>a</sup> and is the updated version of WReSTT (*Web-based Repository of Software Testing Tutorials*) [7].

### 3.3 Software Testing Course

The undergraduate software testing class at Florida International University (FIU), CEN4072 Fundamentals of Software Testing, is offered during the fall, spring and summer semesters. The catalog description of the courses includes: test plan creation, test case generation, program inspections, black-box testing, white-box testing, GUI testing, and the use of testing tools. The prerequisite for CEN4072 is the data structures course. The grading policy for the course is based on 2 midterm exams (25 % each), a group project (25%), attendance and class participation (5%) and a final exam (20%). The course is structured as a lecture section with no lab component, and students are expected to work on the project outside of class on their own time, with some in class question sessions. The textbook currently used in the course is “Foundations of Software Testing” by Mathur [21]. Other reading material includes class notes and tutorials on testing tools.

The topics covered in the CEN4072 course are centered around black-box and white-box testing techniques [21]. The black-box testing techniques, sometimes referred to as *specification-based* testing techniques, presented in the course include equivalence partition, boundary-value analysis, and state-based testing. The white-box techniques, sometimes referred to as *implementation-based* testing techniques, include control flow coverage (statement, branch, multiple condition, and basis path) and data flow coverage (all definitions, all uses, and all definition-use paths). The objective of the course project is to provide students with the experience of analyzing and validating software, writing test cases, creating test documents, and working in teams. The software to be tested during the project is a software application written by a previous software engineering class (undergraduate or graduate). For additional details on the software testing course the interested reader can peruse Clarke et al. [9].

### 3.4 Integrating LESs in a Software Testing Class

In this section we describe how LESs are integrated into the CEN4072 class in both the F2F activities and the online activities using SEP-CyLE. The CEN4072 class is taught in an active learning classroom which contains 5 whiteboards and a wireless connection to the instructor’s projector.

**(1) F2F Activities:** The LESs used in the F2F activities are implemented using the following approach:

- *Collaborative Learning (CL)* - occurs both at the class level and at the project team level. At the class level, students present solutions and answer questions during a class session that may benefit the entire class, e.g., how to run a software testing tool. In addition, class project presentations are considered collaborative learning at the class level. The team level involves team members solving problems in class using the whiteboards and working on the class project (note that in general for this type of class activity the class time spent on CL is split with PBL and and SI). The students are required to keep a diary of all team meetings related to the class project. In addition, the students in a project team are required to rotate

---

<sup>a</sup><https://stem-cyle.cis.fiu.edu/>

the managerial roles (team leader, minute taker and time keeper) of the project between themselves for the two phases of the project.

- *Gamification (GA)* - during the lectures students are awarded midterm exam points both individually and at the team level if they answer questions correctly posed by the instructor. For example, a student may be rewarded with one midterm point if they can correctly define the term *software testing*. Each team member may be awarded with one midterm point if the team can collectively draw a correct flow graph for a method. If a student finds an error with the solution presented by another team, that student gets a point and the team gets no points. Note there is a maximum of 10 midterm points a student can use towards each midterm, and each student in the class is provided an opportunity to answer a question for points.
- *Problem-based Learning (PBL)* - involves students working in their teams to solve problems as in-class activities (see CL and GA above), or working on the project in class which is complex enough to mimic the type of project they will encounter in industry (usually open-ended, e.g., when to stop testing the system).
- *Social Interaction (SI)* - students interact socially in class by working on the problems provided by the instructor as in-class activities, and conducting short team meetings in class. During the in-class activities all students are required to evaluate the correctness of the solutions placed on the whiteboards by a team (see GA above). Note that although the correction of solutions on the whiteboards does involve CL and PBL it tends to be more of a SI activity, since there is more interaction between the other students in the class and the instructor.

Since there are three exams in the course, students are given the opportunity to earn midterm points before each of the exams. If a student earns more than 10 midterm points the extra points (number of point beyond the limit of 10) are *not* carried forward to the next exam.

**(2) Online Activities:** The online activities are centered on using the features in SEP-CyLE and involve some combination of the LESs: CL, GA and SI. As mentioned in Section 3.2 we are working to integrate virtual problem-based environments into SEP-CyLE for PBL. These LESs are implemented using the following approach:

- *CL* - students are placed in virtual teams (usually same as the F2F project teams) and are required to complete digital learning objects (DLOs) or tutorials individually, but are rewarded as a team (see GA below). Students may also post learning content in SEP-CyLE or other learning management system that benefit the entire class.
- *GA* - students are rewarded with virtual points based on the completion of various learning activities (DLOs or tutorials) both individually and as a team (assuming CL is enabled). Below is an example of how points may be awarded:
  - Each student receives: 10 points if she/he completes a DLO and scores 70% or above on the graded assessment quiz; 2 points for writing on the discussion board, and 1 point for uploading a picture.
  - Each member of a team receives: 5 points if all members of the team scores 70% or higher on a DLO; 10 bonus points for the first team to complete a DLO with all team members scoring 70% or higher; 5 bonus points for the second team; and 3 bonus points for the third team.

Table 1: Student data including mean scores and standard deviation (M (SD)) on Midterm Exams 1 (MT1) and 2 (MT2), and extra points earned toward midterm exams.

Group IDs	Class	N	Student Grades (%)		Extra Points	
			MT1* M (SD)	MT2* M (SD)	MT1 <sup>+</sup> M (SD)	MT2 <sup>+</sup> M (SD)
Control (MinLESSs)	CEN4072 - FA18	20	80.5 (7.7)	73.0 (10.6)	3.3 (1.8)	2.3 (1.7)
Treatment (LESSs)	CEN4072 - SP18	19	78.8 (11.0)	76.1 (13.0)	3.4 (3.7)	5.2 (3.6)

MT1\* and MT2\* do not include the extra points earned during the class activities.  
 MT1<sup>+</sup> extra points earned before MT1, and MT2<sup>+</sup> extra points earned between MT1 and MT2.

- There is a leader board showing the top 5 students in the class and each student can see their progress on each assigned DLO and tutorial. We recommend using the anonymized option in SEP-CyLE to mask the student names if the virtual points will count towards the course grade.
- *SI* - students post learning content that help other students in the class, rate and comment on the contents of a DLO and tutorial, and upload a picture. We are still working on a rating system to automatically determine how valuable a post is to the class.

#### 4 Experimental Study

The main objective of our study is to determine if increasing the use of LESSs in the F2F activities of the software testing class improves student learning as reflected in the student grades obtained in the midterm exams. In this section, we present the research method used, the results, and a discussion of the results.

##### 4.1 Method

*Sample:* Students from two (2) *CEN4072 Fundamentals of Software Testing* classes participated in the study, see the three leftmost columns of Table 1. The control group included 20 students in the CEN4072 Fall 2018 class who were minimally exposed to LESSs between MT1 and MT2 in F2F class activities, see Row 2 in Table 1 with entry *Control (MinLESSs)* in Column 1. The treatment group included 19 students in the CEN4072 Spring 2018 class who were fully exposed to LESSs between MT1 and MT2 (LESSs), see Row 3 in Table 1 with entry *Treatment (LESSs)* in Column 1.

*Data Collection:* The data collected includes the scores on the midterm exams (MTs) and the extra points awarded in the class before each MT, see Table 1. The columns in the table are as follows: Column 1 shows the groups, control (MinLESSs) and treatment (LESSs); Column 2 the classes using the pedagogical approach (MinLESSs or LESSs); Column 3 the total enrollments for the classes in each group; Columns 4 and 5 the average grades for each group’s midterm exam (MT1 and MT2), prior to adding the extra points; and the Columns 6 and 7 the extra points earned before MT1, and between MT1 and MT2, respectively.

The MT grades shown in Table 1 *do not* include the points students received during the in-class F2F activities. The points earned by the students were recorded manually by the instructor and added to the MTs after the exam. Over the semesters listed in Table 1, MT1 and MT2 assess the

Table 2: Approximate class times (in minutes) and percentages dedicated to each LES for the class interval between MT1 and MT2. The total class time for this interval was 750 minutes.

Pedagogical Approach	Control (MinLESs)		Treatment (LESs)	
	Time (mins.)	Percentage	Time (mins.)	Percentage
Lecture Style (LS)	380	51%	165	22%
Collaborative Learning (CL)	185	25%	235	31%
Gamification (GA)	60	7%	90	12%
Problem-based Learning (PBL)	50	7%	180	24%
Social Interaction (SI)	75	10%	80	11%

students' knowledge on different topics in the CEN4072 course, however across semesters MT1 assesses similar software testing topics, similarly for MT2. For example, the first question on MT1 is always to define various software testing terms, e.g., definition of *software testing*. The structure of MT2 is as follows: Q1 - define several terms; Q2 - draw a diagram to show the testing process for the project or explain additional terms; Q3 - draw a flow graph for a method, identify test inputs for different code coverage criteria; Q4 - identify definition-use pairs for variables in the method in Q3; and Q5 - strong and weak mutation testing. For this study the MT2s in Fall 2018 and Spring 2018 were identical. Data was also collected in SEP-CyLE [7], however that data is not used in this study.

*Design:* We use a quasi-experimental posttest-only design [22] to determine the impact of using MinLESs versus full LESs in the software testing class. The posttest measures used in the study are the grades of the MT2 exam. Although there is no pretest used in this type of design, we include the scores of MT1 to gauge the abilities of the students in the two software testing class groups at the start of the study.

The intervention in the study is the exposure of the students in the treatment group to higher percentages of LESs and a lower percentage of the traditional lecture style approach when teaching a class. Table 2 shows the approximate class times and percentages dedicated to each LES and the traditional lecture style during the study. The study was conducted during the interval of the semester after MT1 and before MT2 that consisted of 750 minutes total meeting time. These class times in the table were computed based on the number of slides presented during class and the lesson plan for each class. Examples of the class times for Week 7 (first week of the study) for the control and treatment groups are as follows: *Control*, Lecture Style (LS) - 100(67%), Collaborative Learning (CL) - 15(10%), Gamification (GA) - 15(10%), Problem-based learning (PBL) - 10(6.5%), and Social Interaction (SI) - 10(6.5%); and *Treatment*, LS - 60(40%), CL - 20(13%), GA - 30(20%), PBL - 30(20%), and SI - 10(7%). While the control group remained relatively constant with respect to class times for LESs and Lecture Style during the study, in Week 11 (last week of the study) the times for the treatment group were as follows: LS - 10(6.5%), CL - 35(23%), GA - 20(13%), PBL - 65(43.5%), and SI - 20(14%) (this was the MT2 review class session).

The LESs are defined in Section 2.1 and a description of how they are integrated in the class activities is in Section 3.4. The description of the integration of LESs into class activities do involve some overlap, e.g., students working on solving problems in class and working on the

class project can be considered as CL, PBL and SI. In the event where there is a significant overlap of LESs we allocate equal class time to each of the LESs in Table 2. The main differences between the control group and the treatment group are as follows.

1. The number of problems students work on as part of the in-class activities. These problems include both problems that focus on the basic concepts of software testing and those related to the class project. In the treatment group students collaboratively work on these problems thereby increasing the class time for CL, PBL and SI. While in the control group the problems are worked on by the instructor using the traditional lecture style approach thereby maintaining a high percentage of class time for LS.
2. The interval between MT1 and MT2 includes the project's first presentation which consists mainly of CL and SI. Without the first presentation the percentages of class time dedicated to the LESs and traditional lecture style would be: Control group, LS - 63%, CL - 11%, GA - 10%, PBL - 8%, and SI - 8%; and for the Treatment group, LS - 28%, CL - 19%, GA - 15%, PBL - 30%, and SI - 8%, as compared to the values shown in Table 2.

Based on the results of a preliminary study conducted in prior semesters, students in the control group (MinLESs) were given an extra credit question providing them with an opportunity to gain extra MT points. These extra points were used to make up for any points the students may have lost due to the pedagogical approach used (MinLESs). As previously stated both the control and treatment groups were exposed to SEP-CyLE which impacted the online activities in the classes. These online activities were *exactly the same* and included the assignment of 10 DLOs and 3 Tutorials during the course of the semester. The LESs used in SEP-CyLE were GA and SI, CL was turned off. Since the conditions for using SEP-CyLE were the same in both groups we do not expect that using LESs in online class activities impacted the results of the study.

## 4.2 Results and Analysis

Table 1 shows a summary of the data collected including: the group ids, classes participating in the study, the number of participants in each group (N), the mean and standard deviation (M(SD)) of the student scores for the two midterms (MT1\* and MT2\*), and the mean and standard deviation for extra points awarded in class. The results in Table 1 show that in both groups the scores on MT2 were lower than MT1. Note that the midterm exams (MT1 and MT2) were two different exams that are testing two completely different sets of topics. Therefore we cannot use the scores in MT1 and MT2 to measure knowledge gain, hence the posttest only design. The MT1s for both groups were equivalent at the question level (the questions on each test were matched to test the same concepts and course content) with the exception of one area (See Section 4.3, Threats to Validity). The MT2s were exactly the same.

The initial view of the results would tend to support the possibility that the decrease in LS (Lecture Style) class time and increase in the use of other LESs (see Table 2), given the context in Section 3.4, improves student learning based on the MT2 scores only. The table also shows that the in-class points earned by the students in both groups before MT1 are about the same, and the points earned between MT1 and MT2 is more for the treatment group (5.2) than the control group (2.3). This is expected due to the reduced opportunity of students in the control group to earn extra points.

A more in-depth statistical analysis was conducted and the results are as follows. To determine the

statistical test to be performed on the MT2 scores we conducted a Shapiro-Wilk test for normality, and it was found that the MT2 scores are not normally distributed. To determine if the MT2 score between the control and treatment groups were statistically significant we conducted Mann-Whitney U test. The MT2 scores of the control (MinLESs) group ( $Mdn = 76.0$ ) did not differ significantly from the treatment (LESs) group ( $Mdn = 80.0$ ) where  $U = 234$ ,  $z = 1.24$ ,  $p = 0.224$ ,  $r = 0.20$ . Note that the mean ranks for the treatment group is 22.32 and for the control group is 17.70, which also suggests that the treatment group performed better than the control group, although the difference was not significant. A statistical analysis of the MT1 scores prior to administering the treatment was performed to see if there was a statistically significant difference between the 2 groups. Unlike the MT2 test scores, the MT1 scores were normally distributed so we computed one way ANOVA. The results were  $F(1,37) = 0.301$ ,  $p > 0.05$ , which *did not* indicate a significant difference between the groups.

### 4.3 Discussion

The main objective of our study was to investigate the impact on learning gains by increasing the use of LESs in F2F class activities in a software testing class. Specifically, we wanted to determine whether this approach would improve student learning as reflected in the grades students obtained on the second midterm exam (MT2). Our results indicate that the addition of LESs in F2F activities can potentially positively impact student learning gains as reflected by the MT2 scores shown in Table 1 in the column labeled MT2. This assertion is further supported by the fact that students in the control group did better on the MT1 exam, possibly hinting that they may have slightly more ability than those in the treatment group. However, the change shown in the mean scores for MT2 was *not* significant and would infer that additional studies with larger sample sizes are needed. This assumption is based on the statistical analysis conducted on the data collected in a preliminary study that motivated this study. We do not present the preliminary study results due to the fact that there was a four year difference between the data collection of the control group and that of the treatment group. During this 4-year period additional external software testing resources became available to students and would be a threat to validity of the study.

Although the results of this study do not show a statistically significant effect on student learning, it appears that the more F2F class time spent engaging in LES activities, the greater the impact on student learning based on the data in Table 2. Whereas the MinLESs group engaged in a mainly traditional lecture style approach for 51% of the time, the experimental LESs group increasingly engaged in the other pedagogical approaches and less in the traditional lecture style (22%). If the results of our study were significant, using the the proposed model in Figure 1, the integration approach described in Section 3.4 (F2F Activities) and the percentages in Table 2, we would be able to infer that in the software testing class at FIU using the LESIM with the form  $F2FA(0.31CL + 0.12GA + 0.24PBL + 0.11SI + 0.22LS)$  has a greater effect on student learning than  $F2FA(0.25CL + 0.07GA + 0.07PBL + 0.10SI + 0.51LS)$ .

Since the results in our study are not conclusive we plan to conduct additional studies to investigate how increasing LESs may impact both student learning and engagement. In our subsequent studies we will increase LESs systematically to different levels in order to determine the parameters of the LESIM that will most increase student learning. We hope that researchers will use the proposed LESIM, along with an integration approach (see Section 3.4), to identify the parameters for LESIM

that improves student learning and engagement in both F2F and online class activities.

*Threats to Validity.* Threats that may have influenced the results of the study are as follows. The MT1 exams for the two groups were slightly different. A review of the MT1 exams for the two classes differed only in one definition, and the problem domain for a question to draw a state machine and identify test cases from the state machine. Students in both classes were exposed to online activities during the period of the study, the online activities were very similar for both groups thereby reducing this threat. These online activities included the assignment of exactly the same DLOs, tutorials and online LESs using SEP-CyLE. One aspect of the study that needs refinement is using a technique that can more accurately capture the class time dedicated to LESs for the class activities, currently the percentages given are fairly close estimates. Another aspect of the study that needs additional refinement is the allocation of class time to LESs.

## 5 Conclusion and Future Work

In this paper we propose a model for integrating learning and engagement strategies (LESs) into face-to-face (F2F) and online class activities, with the expectation of improving student learning and engagement. These LESs include *collaborative learning*, *gamification*, *problem-based learning* and *social interaction*. The online class activities are supported by a cyberlearning environment (SEP-CyLE) that provides vetted learning content using embedded LESs. We report on a preliminary study that shows increasing LESs in F2F class activities and reducing the traditional lecture style time can potentially improve student learning. In the future we plan to conduct additional studies with larger sample sizes to show our results are significant with respect to student learning and engagement, and investigate how best to integrate LESs into F2F and online class activities.

## Acknowledgment

The authors would like to thank the reviewers for their thoughtful and helpful comments. Partial support for this work was provided by the National Science Foundation's Improving Undergraduate STEM Education (IUSE) program under Award Numbers DUE-1225742 and DUE-1525112 (FIU), DUE-1525555 (FAMU), and DUE-1562773 (FGCU). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- [1] Computing Research Association. Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006, 2017. <https://cra.org/data/Generation-CS/> (retrieved August, 2018).
- [2] Kelsey Hood Cattaneo. Telling active learning pedagogies apart: from theory to practice. *Journal of New Approaches in Educational Research*, 6(2):144–152, 2017.
- [3] Judy Kay, Michael Barg, Alan Fekete, Tony Greening, Owen Hollands, Jeffrey H. Kingston, and Kate Crawford. Problem-based learning for foundation computer science courses. *Computer Science Education*, 10(2):109–128, 2000.
- [4] Edward F. Gehringer. Active and collaborative learning strategies for teaching computing. In *Proceedings of the 2007 ASEE Annual Conference & Exposition*. ASEE, 2007. Paper No. 2328.

- [5] Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3):75–88, 2015.
- [6] Beth Hurst, Randall Wallace, and Sarah B. Nixon. The impact of social interaction on student learning. *Reading Horizons*, 52(4):375–398, 2013. [https://scholarworks.wmich.edu/reading\\_horizons/vol152/iss4/5](https://scholarworks.wmich.edu/reading_horizons/vol152/iss4/5) (retrieved August 2018).
- [7] Raymond Chang-lau and Peter J. Clarke. Software engineering and programming cyberlearning environment (SEP-CyLE), July 2018. <https://stem-cyle.cis.fiu.edu/instances>.
- [8] Peter J. Clarke, Debra L. Davis, Raymond Chang-Lau, James Kiper, Yujian Fu, and Gursimran S. Walia. Using WReSTT cyberlearning environment in the classroom. In *124th American Society for Engineering Education (ASEE)*. ASEE, 2017. Paper No. 19953.
- [9] Peter J. Clarke, Debra L. Davis, Raymond Chang-Lau, and Tariq M. King. Impact of using tools in an undergraduate software testing course supported by WReSTT. *ACM Trans. Comput. Educ.*, 17(4):18:1–18:28, August 2017.
- [10] Michael Prince. Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3):223–231, 2004.
- [11] Barbara Leigh Smith and Jean T. MacGregor. What is Collaborative Learning? In A.S. Goodsell, M.R. Mather, and V. Tinto, editors, *Collaborative Learning: A Sourcebook for Higher Education*. National Center on Postsecondary Teaching, Learning, and Assessment, University Park, Pa., 1992.
- [12] Thomas W. Malone. What makes things fun to learn? heuristics for designing instructional computer games. In *Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems*, SIGSMALL '80, pages 162–169, New York, NY, USA, 1980. ACM.
- [13] H. G. Schmidt. Foundations of problem-based learning: some explanatory notes. *Medical Education*, 27(5):422–432, 1993.
- [14] Sandra Y. Okita. *Social Interactions and Learning*, pages 3104–3107. Springer US, Boston, MA, 2012.
- [15] Ilaria Liccardi, Asma Ounnas, Reena Pau, Elizabeth Massey, Päivi Kinnunen, Sarah Lewthwaite, Marie-Anne Midy, and Chandan Sarkar. The role of social networks in students' learning experiences. *SIGCSE Bull.*, 39(4):224–237, December 2007.
- [16] Liu Junhua, Yue Zhang, Justin Ruths, Diana Moreno, Daniel D. Jensen, and Kristin L. Wood. Innovations in software engineering education: An experimental study of integrating active learning and design-based learning. In *Proceedings of the 120th ASEE Annual Conference & Exposition*. ASEE, 2013. Paper No. 6451.
- [17] Paul Chandler and John Sweller. Cognitive load theory and the format of instruction. *Cognition and Instruction*, 8(4):293–332, 1991.
- [18] Ruth Colvin Clark, Frank Nguyen, John Sweller, and Melissa Baddeley. Efficiency in learning: Evidence-based guidelines to manage cognitive load. *Performance Improvement*, 45(9):46–47, 2006.
- [19] Michael S. Franklin, Jonathan Smallwood, Claire M. Zedelius, James M. Broadway, and Jonathan W. Schooler. Unaware yet reliant on attention: Experience sampling reveals that mind-wandering impedes implicit learning. *Psychonomic Bulletin & Review*, 23(1):223–229, Feb 2016.
- [20] Rachel S. Smith. Guidelines for authors of learning objects. The New Media Consortium, 2004.
- [21] Aditya P. Mathur. *Foundations of Software Testing*. Addison-Wesley Professional, 2nd edition, 2014.
- [22] John W. Creswell. *Educational research: Planning, conducting, and evaluating quantitative and qualitative research*. Prentice Hall Upper Saddle River, NJ, 4th edition, 2014.